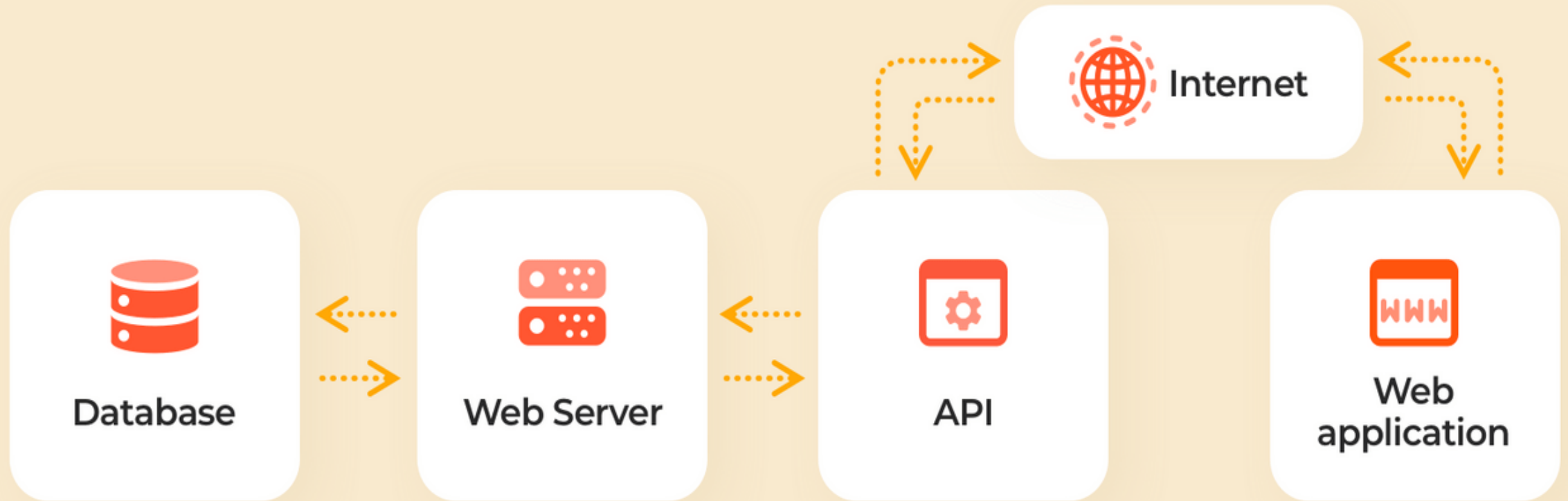


# Les API

Définition, utilité, fonctionnement...

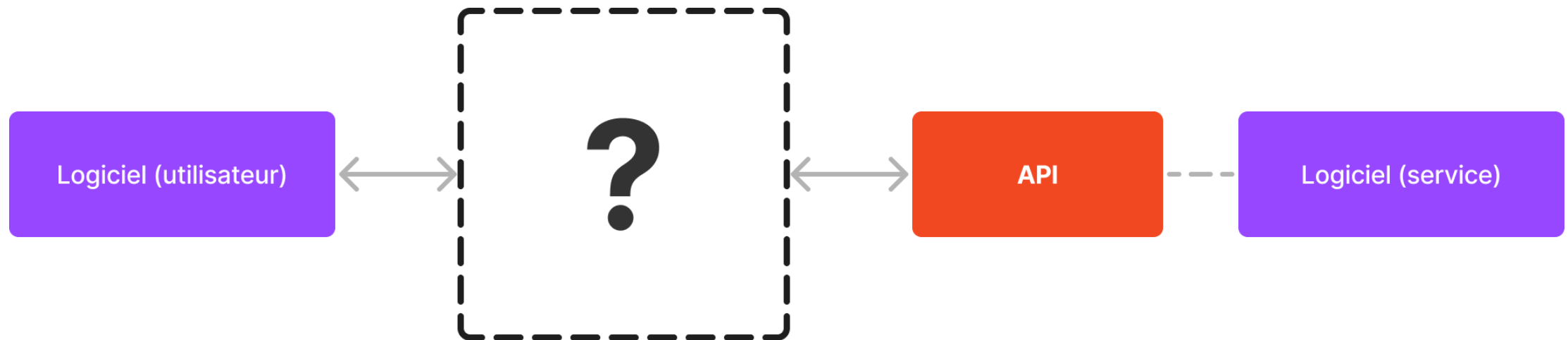
# ~~What is an API?~~



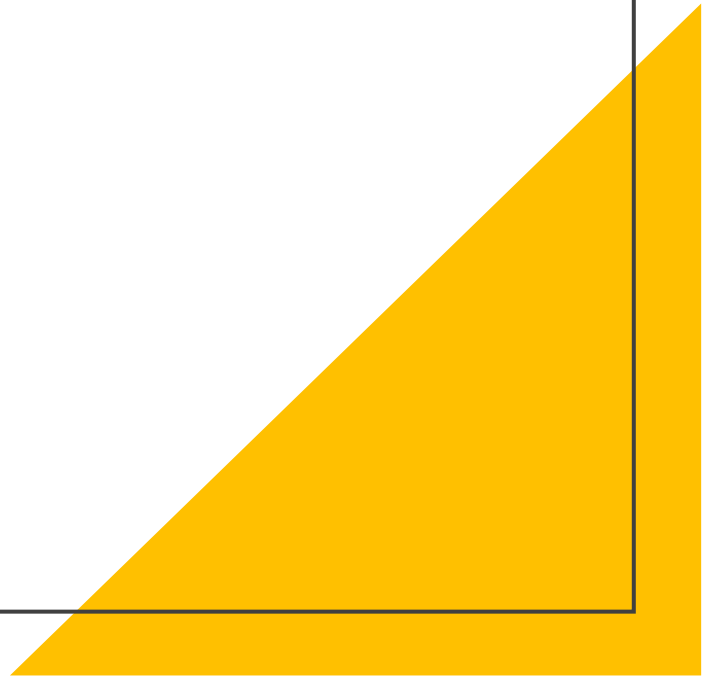
# En une phrase...

Une interface de programmation d'application (API pour **Application Programming Interface**) est un ensemble normalisé de classes, de méthodes, de fonctions et de constantes qui sert de **façade** par laquelle un **logiciel offre des services à d'autres logiciels**.

[https://fr.wikipedia.org/wiki/Interface\\_de\\_programmation](https://fr.wikipedia.org/wiki/Interface_de_programmation)



# Des exemples d'API ?



# Quand les utiliser ?

**Presque tout le temps en réalité...**

**L'API est la conséquence logique d'un besoin d'exposition de services.**

**Par exemple...**

- 1. Je modularise le code (bonne pratique)**
- 2. Les modules ont besoin d'un moyen pour communiquer**
- 3. On définit une API (contrat) à cet effet**

# Les différents usages

Interne/externe, distant/local...

# Interne/externe

Une API n'est pas forcément destinée à être utilisée par un public externe.

Une entreprise peut par exemple décider de réserver l'usage d'une API (REST, ...) à ses employés uniquement.

À l'inverse, une entreprise peut décider d'exposer des services via une API externe (ex. Spotify).

# Distant/local

Une API peut être destinée à être utilisée **localement** sur une et **même machine physique** (bibliothèque, socket, etc.) ou de façon **distance** (via TCP/IP, etc.).



# Les types d'architecture

API distantes

# RPC (Remote Procedure Calls)

- Protocole permettant l'appel de procédure sur un ordinateur distant
  - Architecture client/serveur
  - Beaucoup de sous-protocoles mais tous incompatibles...
- 
1. Le client envoie une requête d'invocation de procédure au serveur
  2. Le serveur exécute la requête et invoque la procédure
  3. Le serveur envoie la réponse au client

# SOAP, un exemple de RPC

- **Simple Object Access Protocol**
- **Protocol d'échange d'information structurée**
- **Utilise XML pour le format des données et HTTP pour le transfert d'information**

# Requête SOAP

[https://www.w3schools.com/xml/xml\\_soap.asp](https://www.w3schools.com/xml/xml_soap.asp)

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
```

```
<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

  <soap:Body xmlns:m="http://www.example.org/stock">
    <m:GetStockPrice>
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```

# Réponse SOAP

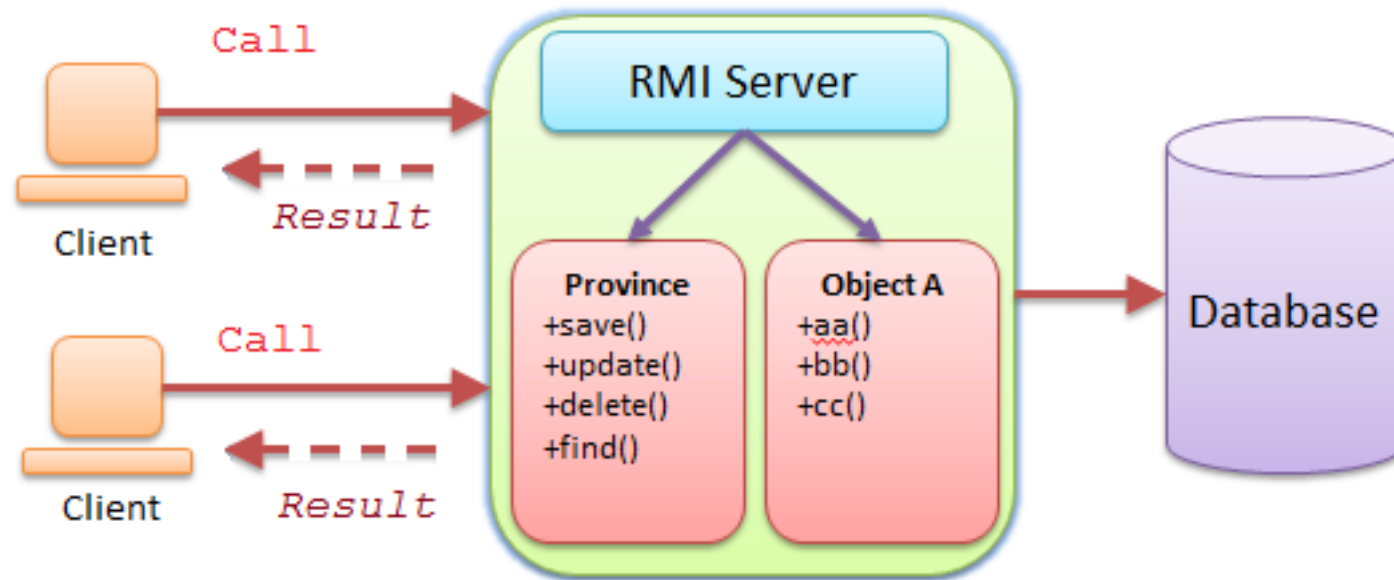
[https://www.w3schools.com/xml/xml\\_soap.asp](https://www.w3schools.com/xml/xml_soap.asp)

HTTP/1.1 200 OK  
Content-Type: application/soap+xml; charset=utf-8  
Content-Length: nnn

```
<?xml version="1.0"?>  
  
<soap:Envelope  
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"  
  soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">  
  
  <soap:Body xmlns:m="http://www.example.org/stock">  
    <m:GetStockPriceResponse>  
      <m:Price>34.5</m:Price>  
    </m:GetStockPriceResponse>  
  </soap:Body>  
  
</soap:Envelope>
```

# Java RMI (Remote Method Invocation)

- Complètement incompatible avec RPC mais fait la même chose
- Orienté objet



# REST

- **Respresentational State Transfer**
- Orienté **ressource**
- Utilise HTTP pour le transfert de données
- N'impose pas de format de données

# Points clés à retenir

- Une API ne définit pas une implémentation mais un **contrat**.
- Il existe **plusieurs architectures** d'API différentes ! Et pas seulement REST...
- Une API n'est pas forcément destinée à être exposée au public.



# Focus sur les API REST

- Très flexible
- Mature
- Simple
- Utilisé par la majorité des entreprises, dont les GAFAM